

Claims

What is claimed is:

1. A method of expressing an association between a declaration and an implementation in a language neutral fashion, comprising:
 - determining whether a source language association rule related to a declaration is different from a default association rule for a target runtime;
 - expressing an association between the declaration and an implementation according to an override association rule for the target runtime if the source language association rule is different from the default association rule for the target runtime; and
 - expressing an association between the declaration and the implementation according to the default association rule if the source language association rule is the same as the default association rule for the target runtime.
2. The method of claim 1, wherein expressing an association according to an override association rule comprises expressing an explicit association between the declaration and the implementation.
3. The method of claim 2, wherein expressing the explicit association comprises creating an association between a class, a code body associated with the implementation, and the declaration.
4. The method of claim 3, wherein creating an association between the class, the code body associated with the implementation, and the declaration comprises creating an entry in an override association table having entries for the class, the code body, and the declaration.
5. The method of claim 1, wherein the declaration comprises a method declaration signature, the implementation comprises an implementation signature in a class, the default association rule for a target runtime comprises signature matching, and wherein expressing an association between the declaration and an implementation

according to an override association rule comprises expressing an explicit association between the method declaration signature and the implementation signature if the method declaration signature and the implementation signature do not match.

6. The method of claim 5, wherein expressing the explicit association comprises creating an association between the class, a code body associated with the implementation, and the declaration.

7. The method of claim 4, wherein creating an association between the class, the code body associated with the implementation, and the declaration comprises creating an entry in an override association table having entries for the class, the code body, and the declaration.

8. The method of claim 5, wherein the method declaration signature matches a first implementation signature associated with a first implementation in the class and a second implementation signature associated with a second implementation in the class, and wherein expressing an association between the declaration and an implementation according to an override association rule comprises expressing an explicit association between the method declaration signature and one of the first and second implementation signatures.

9. The method of claim 8, wherein expressing the explicit association comprises creating an association between the class, a code body associated with one of the first and second implementations, and the declaration.

10. The method of claim 9, wherein creating an association between the class, the code body associated with one of the first and second implementations, and the declaration comprises creating an entry in an override association table having entries for the class, the code body, and the declaration.

11. The method of claim 8, further comprising selecting one of the first and second implementations in the class for association with the declaration according to a source language association rule.

12. A method of expressing an association between a declaration and an implementation in a language neutral fashion, comprising:

expressing an association between a source language declaration and an implementation according to an override association rule for a target runtime.

13. The method of claim 12, wherein expressing an association according to an override association rule comprises expressing an explicit association between the declaration and the implementation.

14. The method of claim 13, wherein expressing the explicit association comprises creating an association between a class, a code body associated with the implementation, and the declaration.

15. The method of claim 14, wherein creating an association between the class, the code body associated with the implementation, and the declaration comprises creating an entry in an override association table having entries for the class, the code body, and the declaration.

16. A method of interpreting an association between a declaration and an implementation in a target runtime system, the method comprising:

determining whether the association comprises an override association;
interpreting the association between the declaration and the implementation according to an override association rule for the target runtime if the association comprises an override association; and

interpreting the association between the declaration and the implementation according to a default association rule for the target runtime if the association does not comprise an override association.

17. The method of claim 16, wherein the implementation is related to a class, and wherein determining whether the association comprises an override association comprises determining whether a metadata component comprises an explicit association between the declaration and the implementation.

18. The method of claim 17, wherein determining whether the metadata component comprises an explicit association comprises determining whether the metadata component comprises an association between the class, a code body associated with the implementation, and the declaration.

19. The method of claim 18, wherein determining whether the metadata component comprises an association between the class, a code body associated with the implementation, and the declaration comprises determining whether the metadata component comprises an entry in an override association table having entries for the class, the code body, and the declaration.

20. The method of claim 19, wherein interpreting the association between the declaration and the implementation according to an override association rule comprises associating the code body with the declaration in the runtime system according to the override association table.

21. The method of claim 20, further comprising providing the association between the declaration and the implementation to a JIT compiler via a layout component.

22. The method of claim 16, wherein interpreting the association between the declaration and the implementation according to an override association rule comprises associating a code body associated with the implementation with the declaration in the runtime system.

23. The method of claim 16, further comprising providing the association between the declaration and the implementation to a JIT compiler via a layout component.

24. A method of interpreting an association between a declaration and an implementation in a target runtime system, the method comprising:

interpreting the association between the declaration and the implementation according to an override association rule for the target runtime.

25. The method of claim 24, wherein interpreting the association between the declaration and the implementation according to an override association rule comprises associating a code body associated with the implementation with the declaration in the runtime system.

26. The method of claim 25, further comprising providing the association between the declaration and the implementation to a JIT compiler via a layout component.

27. The method of claim 24, further comprising providing the association between the declaration and the implementation to a JIT compiler via a layout component.

28. A computer-readable medium having computer-executable instructions for:

determining whether a source language association rule related to a declaration is different from a default association rule for a target runtime;

expressing an association between the declaration and an implementation according to an override association rule for the target runtime if the source language association rule is different from the default association rule for the target runtime; and

expressing an association between the declaration and the implementation according to the default association rule if the source language association rule is the same as the default association rule for the target runtime.

29. The computer-readable medium of claim 1, wherein the computer-executable instructions for expressing an association according to an override association rule comprises computer-executable instructions for expressing an explicit association between the declaration and the implementation.

30. The computer-readable medium of claim 2, wherein the computer-executable instructions for expressing the explicit association comprises computer-executable instructions for creating an association between a class, a code body associated with the implementation, and the declaration.

31. The computer-readable medium of claim 3, wherein the computer-executable instructions for creating an association between the class, the code body associated with the implementation, and the declaration comprises computer-executable instructions for creating an entry in an override association table having entries for the class, the code body, and the declaration.

32. The computer-readable medium of claim 28, wherein the declaration comprises a method declaration signature, the implementation comprises an implementation signature in a class, the default association rule for a target runtime comprises signature matching, and wherein the computer-executable instructions for expressing an association between the declaration and an implementation according to an override association rule comprises computer-executable instructions for expressing an

explicit association between the method declaration signature and the implementation signature if the method declaration signature and the implementation signature do not match.

33. The computer-readable medium of claim 32, wherein the method declaration signature matches a first implementation signature associated with a first implementation in the class and a second implementation signature associated with a second implementation in the class, and wherein the computer-executable instructions for expressing an association between the declaration and an implementation according to an override association rule comprises computer-executable instructions for expressing an explicit association between the method declaration signature and one of the first and second implementation signatures.

34. A computer-readable medium for expressing an association between a declaration and an implementation in a language neutral fashion having computer-executable instructions for expressing an association between a source language declaration and an implementation according to an override association rule for a target runtime.

35. A computer-readable medium for interpreting an association between a declaration and an implementation in a target runtime system, the computer-readable medium having computer-executable instructions for:

determining whether the association comprises an override association;
interpreting the association between the declaration and the implementation according to an override association rule for the target runtime if the association comprises an override association; and

interpreting the association between the declaration and the implementation according to a default association rule for the target runtime if the association does not comprise an override association.

36. The computer-readable medium of claim 35, wherein the implementation is related to a class, and wherein the computer-executable instructions for determining whether the association comprises an override association comprises computer-executable instructions for determining whether a metadata component comprises an explicit association between the declaration and the implementation.

37. The computer-readable medium of claim 36, wherein the computer-executable instructions for determining whether the metadata component comprises an explicit association comprises computer-executable instructions for determining whether the metadata component comprises an association between the class, a code body associated with the implementation, and the declaration.

38. The computer-readable medium of claim 37, wherein the computer-executable instructions for determining whether the metadata component comprises an association between the class, a code body associated with the implementation, and the declaration comprises computer-executable instructions for determining whether the metadata component comprises an entry in an override association table having entries for the class, the code body, and the declaration.

39. The computer-readable medium of claim 38, wherein the computer-executable instructions for interpreting the association between the declaration and the implementation according to an override association rule comprises computer-executable instructions for associating the code body with the declaration in the runtime system according to the override association table.

40. A computer-readable medium for interpreting an association between a declaration and an implementation in a target runtime system, the computer-readable medium having computer-executable instructions for interpreting the association between the declaration and the implementation according to an override association rule for the target runtime.

41. A system for expressing an association between a declaration and an implementation in a language neutral fashion, comprising:

a first component adapted to determine whether a source language association rule related to the declaration is different from a default association rule for a target runtime; and

an association expression component adapted to express an association between the declaration and an implementation according to an override association rule for the target runtime if the source language association rule is different from the default association rule for the target runtime, and to express an association between the declaration and the implementation according to the default association rule if the source language association rule is the same as the default association rule for the target runtime.

42. The system of claim 41, wherein the association expression component is adapted to express an explicit association by creating an association between a class, a code body associated with the implementation, and the declaration.

43. The system of claim 42, wherein the association expression component is adapted to create an entry in an override association table having entries for the class, the code body, and the declaration.

44. A system for expressing an association between a declaration and an implementation in a language neutral fashion, comprising an association expression component adapted to express an association between a source language declaration and an implementation according to an override association rule for a target runtime.

45. A system for interpreting an association between a declaration and an implementation in a target runtime system, the method comprising:

a first component adapted to determine whether the association comprises an override association; and

a second component adapted to interpret the association between the declaration and the implementation according to an override association rule for the target runtime if the association comprises an override association, and to interpret the association between the declaration and the implementation according to a default association rule for the target runtime if the association does not comprise an override association.

46. A system for interpreting an association between a declaration and an implementation in a target runtime system, the method comprising:

a layout component adapted to interpret the association between the declaration and the implementation according to an override association rule for the target runtime.

47. A source compiler for generating an intermediate language representation of a source code program, comprising:

an association expression system for expressing an association between a source language declaration and an implementation in a language neutral fashion, the association expression system comprising:

a first component adapted to determine whether a source language association rule related to the source language declaration is different from a default association rule for a target runtime; and

an association expression component adapted to express an association between the source language declaration and the implementation according to an override association rule for the target runtime if the source language association rule is different from the default association rule for the target runtime, the association expression component being further adapted to express an association between the declaration and the implementation according to the default association rule if the source language association rule is the same as the default association rule for the target runtime.

48. A method of expressing an association between a declaration and an implementation allowing expression of associations between different languages with different rules, comprising:

determining whether a source language association rule related to a declaration is different from a default association rule for a target runtime;

expressing an association between the declaration and an implementation according to an override association rule for the target runtime if the source language association rule is different from the default association rule for the target runtime; and

expressing an association between the declaration and the implementation according to the default association rule if the source language association rule is the same as the default association rule for the target runtime.